Knowledge Based Obligations

RUC-ILLC Workshop on Deontic Logic

Eric Pacuit

Stanford University

November 9, 2007

In 1964, a young woman (Kitty Genovese) was stabbed to death near her home in the Kew Gardens section of Queens, New York.

In 1964, a young woman (Kitty Genovese) was stabbed to death near her home in the Kew Gardens section of Queens, New York.

She was attacked only about 30m from her apartment's door numerous times by a man (Winston Moseley) starting around 3:15 a.m.

In 1964, a young woman (Kitty Genovese) was stabbed to death near her home in the Kew Gardens section of Queens, New York.

She was attacked only about 30m from her apartment's door numerous times by a man (Winston Moseley) starting around 3:15 a.m.

The murder took place over a period of about thirty minutes, during which at least 38 alleged "witnesses" failed to help the victim. (Note there is some dispute over what the witnesses *actually* saw and heard)

Why did the neighbors not act to help Kitty?

Why did the neighbors not act to help Kitty?

The reason was not indifference, but none of the neighbours had even a default obligation to act, even though, as a group, they did have an obligation to take some action to protect Kitty.

Why did the neighbors not act to help Kitty?

The reason was not indifference, but none of the neighbours had even a default obligation to act, even though, as a group, they did have an obligation to take some action to protect Kitty.

Issues: obligations, group obligations, knowledge, group knowledge, default obligations, etc.

Why did the neighbors not act to help Kitty?

The reason was not indifference, but none of the neighbours had even a default obligation to act, even though, as a group, they did have an obligation to take some action to protect Kitty.

Issues: obligations, group obligations, knowledge, group knowledge, default obligations, etc.

An agent's obligations are often dependent on what the agent knows, and indeed one cannot reasonably be expected to respond to a problem if one is not aware of its existence.

✓ Introductory Example

2. Motivating Examples

3. Background: History Based Models

4. Knowledge Based Obligations.

   E. Pacuit, R. Parikh and E. Cogan. *The Logic of Knowledge Based Applications*. Knowledge, Rationality and Action (Synthese) 149: 311 - 341 (2006).

5. Discussion

Some Related Work

J. Horty. Agency and Deontic Logic. 2001.

R. van der Meyden. *The Dynamic Logic of Permission*. *Journal of Logic and Computation* **6**:3 (1996).

A. Lomuscio and M. Sergot. *Deontic Interpreted Systems*. *Studia Logica* **75** (2003).

(plus *a lot* of work on deontic logic and epistemic logic)

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.
2. Uma is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Uma's house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.
3. Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary's condition at all times and to provide emergency treatment as appropriate.

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Uma's house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary's condition at all times and to provide emergency treatment as appropriate.

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Uma's house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary's condition at all times and to provide emergency treatment as appropriate.

4. Uma has a patient with a certain condition C who is in the St. Gibson hospital mentioned above. There are two drugs d and d' which can be used for C, but d has a better track record. Uma is about to inject the patient with d, but unknown to Uma, the patient is allergic to d and for this patient d' should be used. Nurse Rebecca is aware of the patient's allergy and also that Uma is about to administer d. It is then Rebecca's obligation to inform Uma and to suggest that drug d' be used in this case.

Four Issues

1. The formal language and semantics must have machinery in which we can express statements of the form "after agent i performs action a...".

Four Issues

1. The formal language and semantics must have machinery in which we can express statements of the form “after agent i performs action a...”.

2. The formal language and semantics must have machinery in which we can express statements of the form “agent i is obliged to perform action a”.

Four Issues

1. The formal language and semantics must have machinery in which we can express statements of the form "after agent i performs action a..."
2. The formal language and semantics must have machinery in which we can express statements of the form "agent i is obliged to perform action a".
3. Certain actions are obligatory *only* in the presence of relevant information.

Four Issues

1. The formal language and semantics must have machinery in which we can express statements of the form "after agent i performs action a...".
2. The formal language and semantics must have machinery in which we can express statements of the form "agent i is obliged to perform action a".
3. Certain actions are obligatory *only* in the presence of relevant information.
4. Certain obligations may disappear in the presence of relevant information: default obligations (example 4).

✓ Introductory Example

✓ Motivating Examples

3. Background: History Based Models

4. Knowledge Based Obligations.

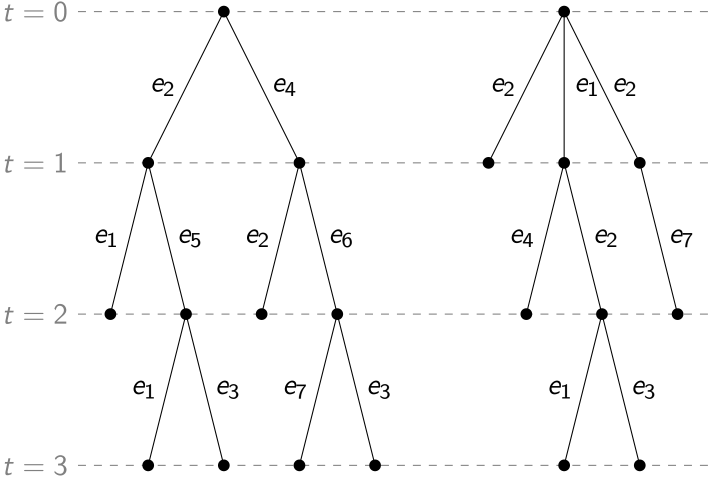
   E. Pacuit, R. Parikh and E. Cogan. *The Logic of Knowledge Based Applications*. Knowledge, Rationality and Action (Synthese) 149: 311 - 341 (2006).

5. Discussion

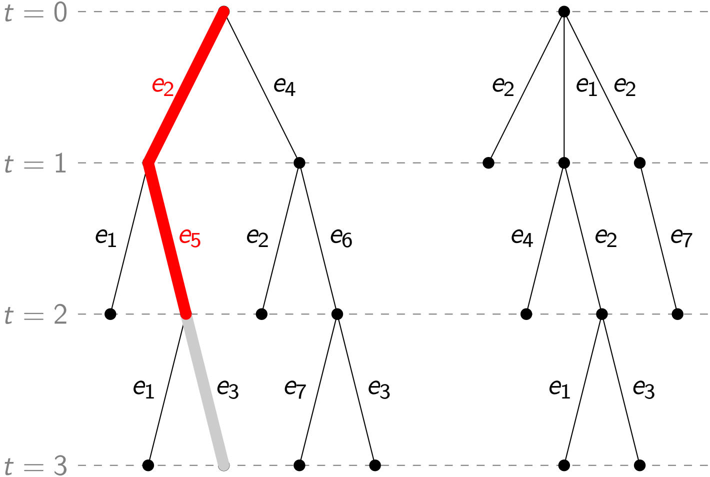
R. Parikh and R. Ramanujam. *A Knowledge Based Semantics of Messages*. *Journal of Logic, Language and Information*, 12: 453 – 467, 2003. (1985).

FHMV. Reasoning about Knowledge. MIT Press, 1995.

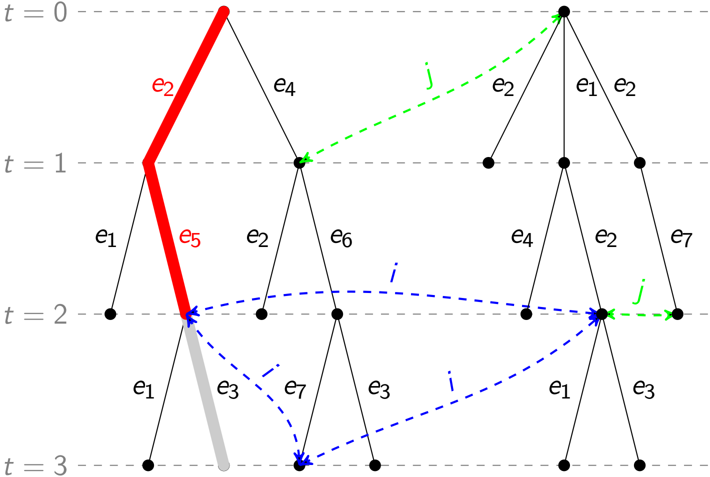
The 'Playground'



The ‘Playground’



The 'Playground'



The ‘Playground’: Notation

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ is the **length** of H.

The ‘Playground’: Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\mathsf{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.
- $\mathsf{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\text{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\text{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\mathsf{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

Definition

An ETL frame is a tuple $\langle \Sigma, \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}} \rangle$ where Σ is a (finite or infinite) set of events, $\mathcal{H}$ is a protocol, and for each $i \in \mathcal{A}$, $\sim_i$ is an equivalence relation on the set of finite strings in $\mathcal{H}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\text{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\text{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

Definition

An ETL frame is a tuple $\langle \Sigma, \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}} \rangle$ where Σ is a (finite or infinite) set of events, $\mathcal{H}$ is a protocol, and for each $i \in \mathcal{A}$, $\sim_i$ is an equivalence relation on the set of finite strings in $\mathcal{H}$.

Some assumptions:

1. If Σ is assumed to be finite, then we say that $\mathcal{F}$ is **finitely branching**.
2. If $\mathcal{H}$ is a rooted protocol, $\mathcal{F}$ is a **tree frame**,.

Defining Uncertainty

Defining Uncertainty

1. Histories or Runs?
   - Let L be a set of **local states.**
   - A **run** $r \in \mathcal{R}$ is a function $r : \mathbb{N} \rightarrow L^{n+1}$
   - Agent i cannot distinguish two points if it is in the same state in both: $(r, t) \sim_i (r', t')$ iff $r(t)_i = r'(t')_i$.

Defining Uncertainty

1. Histories or Runs?
   - Let L be a set of **local states.**
   - A **run** $r \in \mathcal{R}$ is a function $r : \mathbb{N} \rightarrow L^{n+1}$
   - Agent i cannot distinguish two points if it is in the same state in both: $(r, t) \sim_i (r', t')$ iff $r(t)_i = r'(t')_i$.

2. **Local View Functions**: Suppose that $\Sigma = \cup_{i \in \mathcal{A}} E_i$.
   For each $i \in \mathcal{A}$ define $\lambda_i : \mathrm{FinPre}(\mathcal{H}) \rightarrow E_i^*$
   For each **finite** $H \in \mathcal{H}$,

$$H \sim_i H' \text{ iff } \lambda_i(H) = \lambda_i(H')$$

Defining Uncertainty

Characterization Theorem(s) Each approach is “equivalent”.

EP. *Some Comments on History Based Structures*. Journal of Applied Logic, 2007.

Formal Languages

- $P\varphi$ (φ is true *sometime* in the past),
- $F\varphi$ (φ is true *sometime* in the future),
- $Y\varphi$ (φ is true at *the* previous moment),
- $N\varphi$ (φ is true at *the* next moment),
- $N_e\varphi$ (φ is true after event e)
- $K_i\varphi$ (agent i knows φ) and
- $C_B\varphi$ (the group $B \subseteq \mathcal{A}$ commonly knows φ).

History-based Models

An ETL **model** is a structure $\langle \mathcal{H}, \{\sim_i\}_{i\in\mathcal{A}}, V\rangle$ where $\langle \mathcal{H}, \{\sim_i\}_{i\in\mathcal{A}}\rangle$ is an ETL frame and

$V : \mathsf{At} \rightarrow 2^{\mathsf{finite}(\mathcal{H})}$ is a valuation function.

Formulas are interpreted at pairs H, t:

$$H, t \models \varphi$$

Truth in a Model

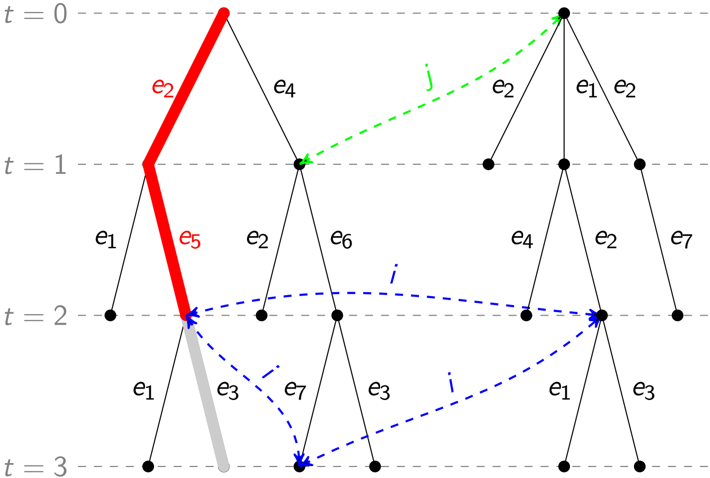
- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
- $H, t \models K_i\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_i H'_m$ then $H', m \models \varphi$
- $H, t \models C\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_* H'_m$ then $H', m \models \varphi$.

where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.

Truth in a Model

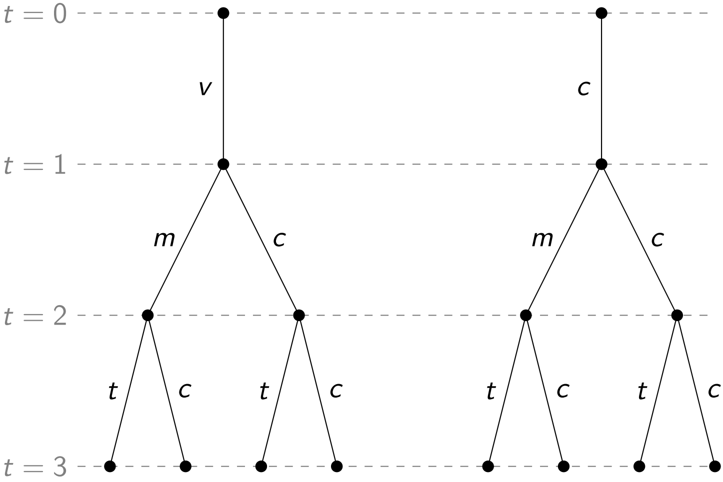
- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
- $H, t \models K_i\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_i H'_m$ then $H', m \models \varphi$
- $H, t \models C\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_* H'_m$ then $H', m \models \varphi$.

where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.

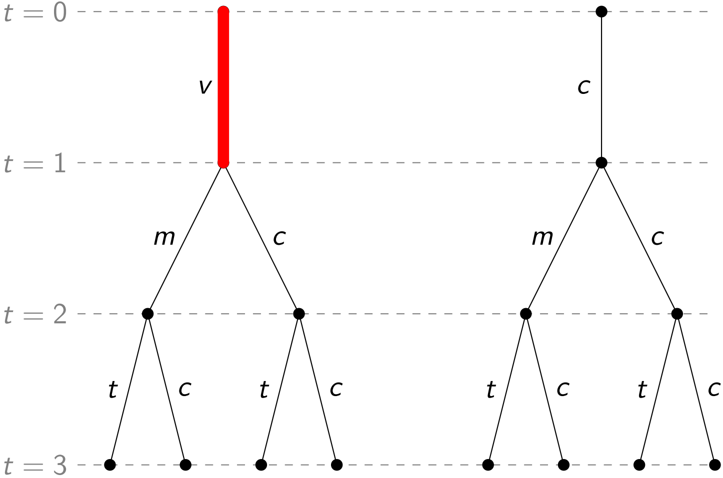


$t = 0$
$t = 1$
$t = 2$
$t = 3$
e_2
e_4
j
e_2
e_1
e_2
e_1
e_5
e_2
e_6
e_4
e_2
e_7
i
e_1
e_3
i
e_7
e_3
i
e_1
e_3

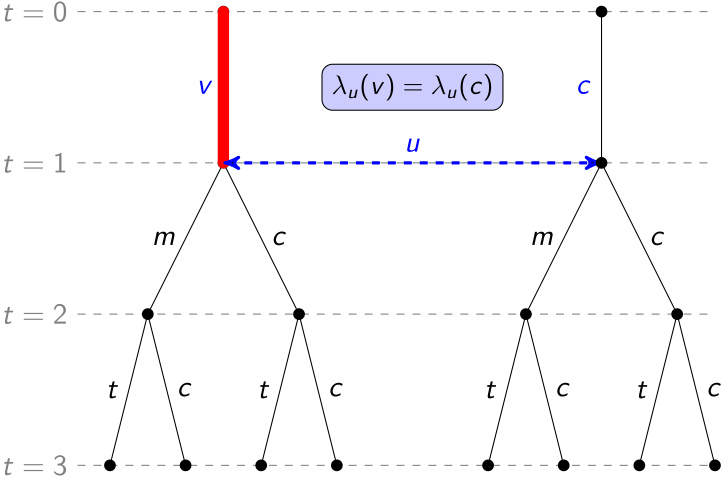
Example 1 & 2



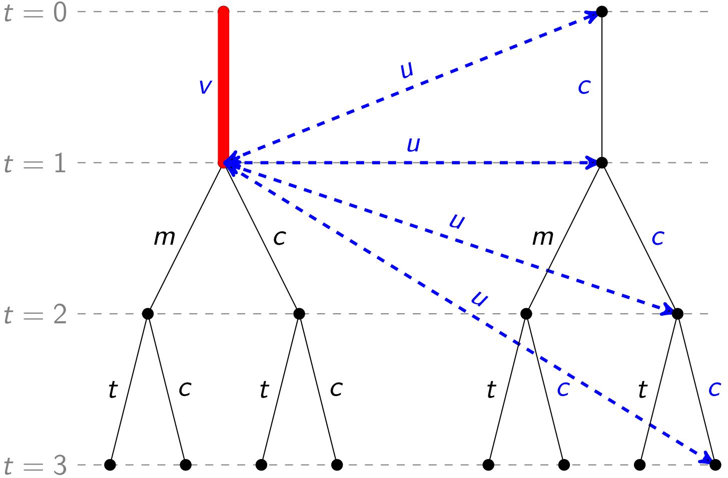
Example 1 & 2



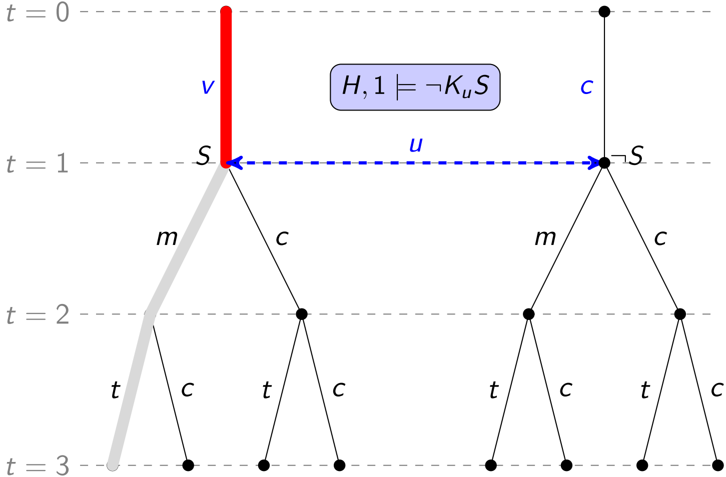
Example 1 & 2



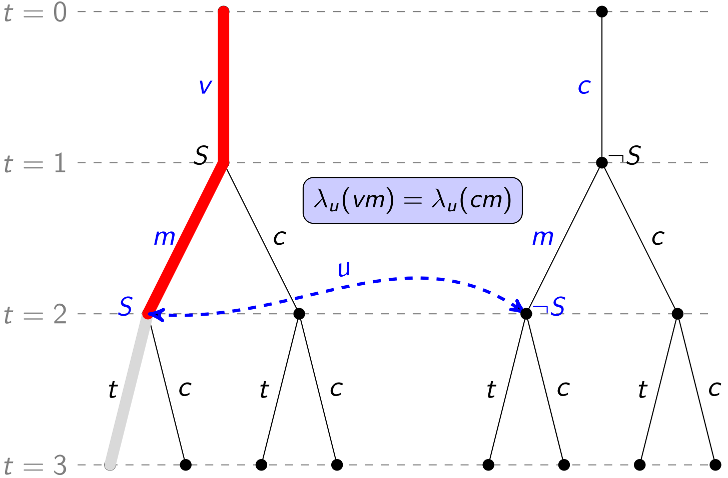
Example 1 & 2



Example 1 & 2



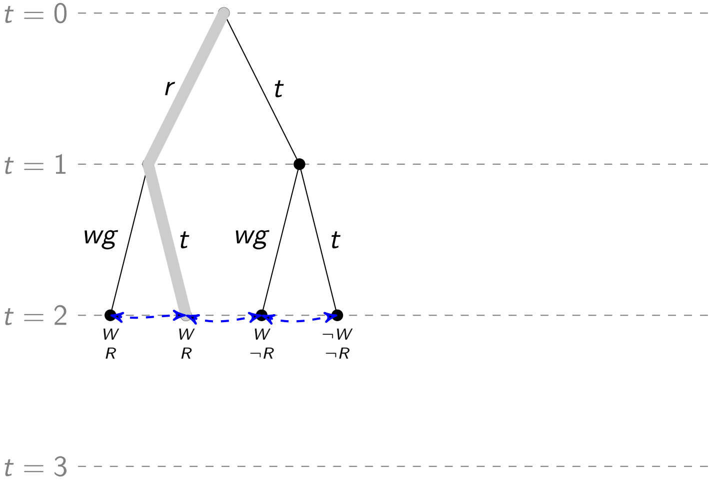
Example 1 & 2



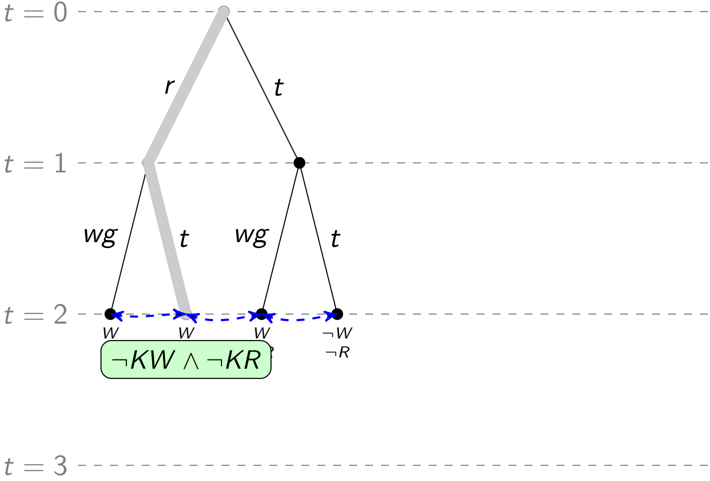
$t = 0$
$t = 1$
$t = 2$
$t = 3$
v
S
m
c
S
u
t
c
t
c
c
$\neg S$
$\lambda_u(vm) = \lambda_u(cm)$
m
c
$\neg S$
t
c
t
c

Learning from the Protocol

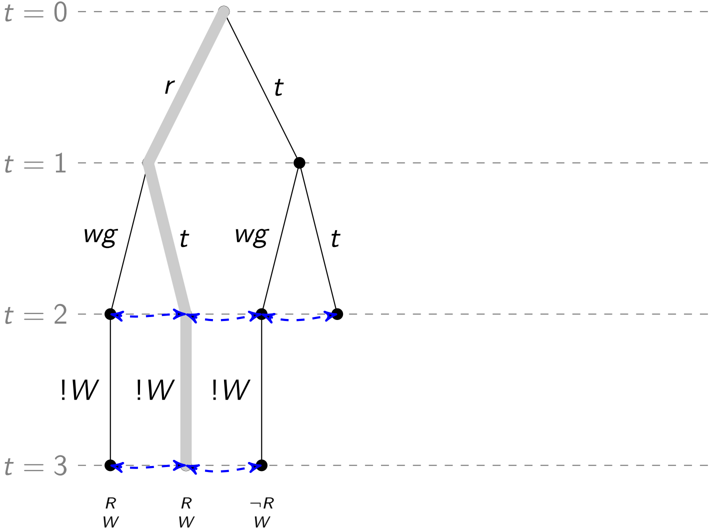
Learning from the Protocol



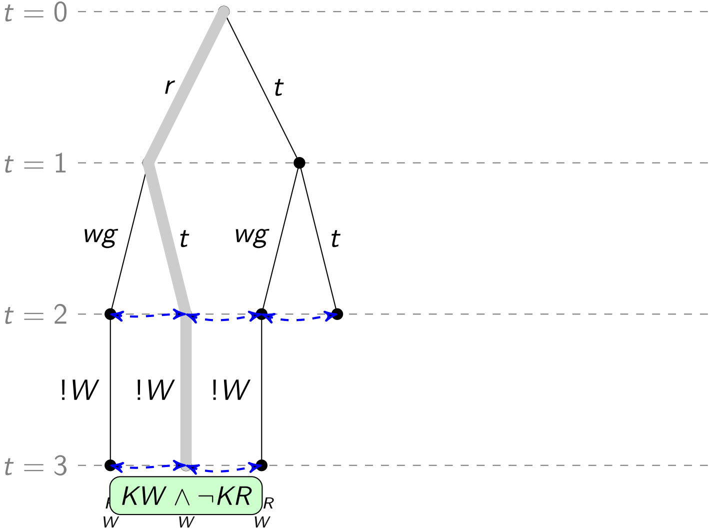
Learning from the Protocol



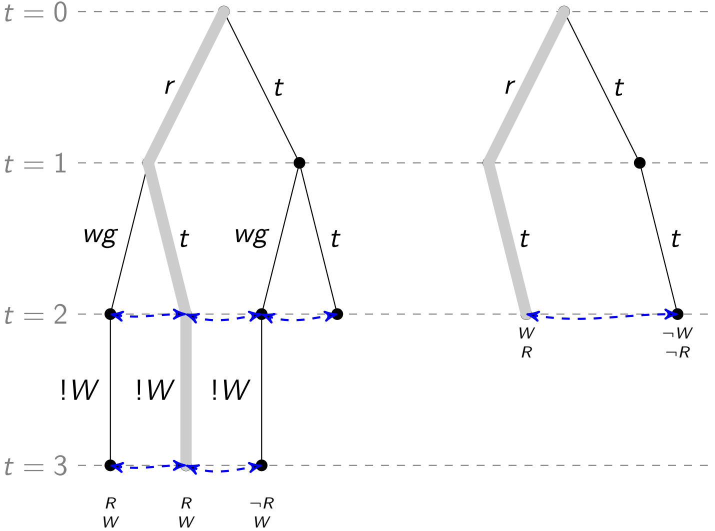
Learning from the Protocol



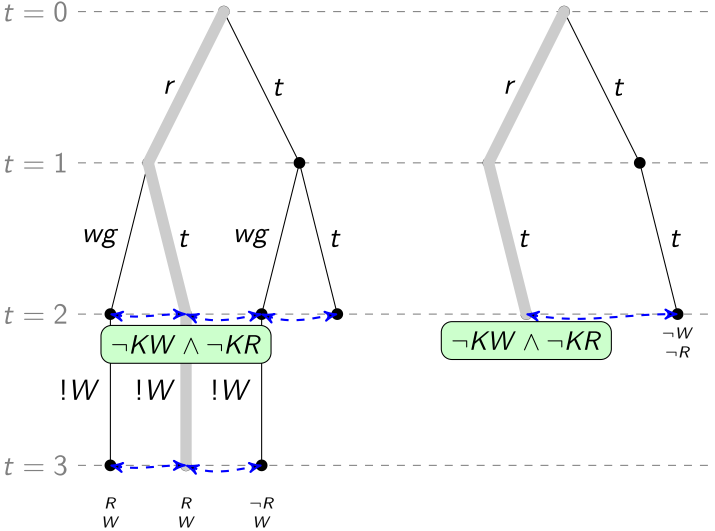
Learning from the Protocol



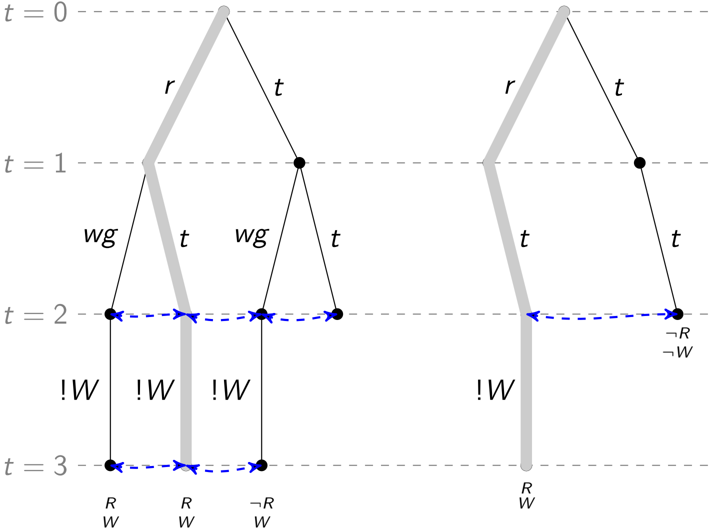
Learning from the Protocol



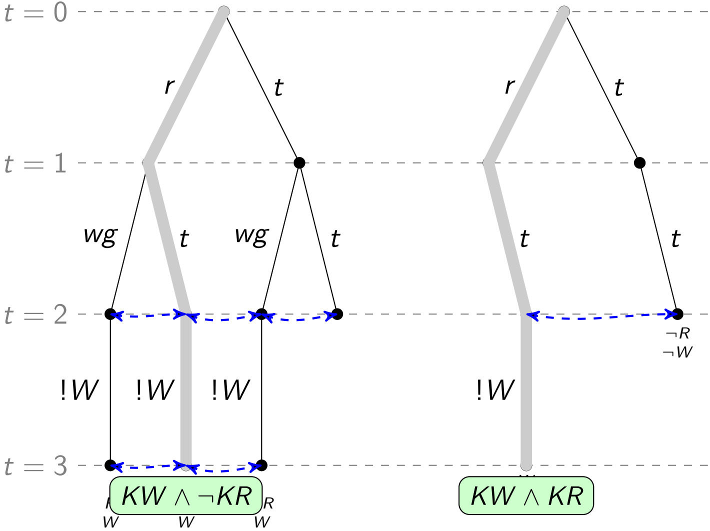
Learning from the Protocol



Learning from the Protocol



Learning from the Protocol



Parameters of the Models

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both? Branching time or linear time?

Parameters of the Models

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both? Branching time or linear time?

2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

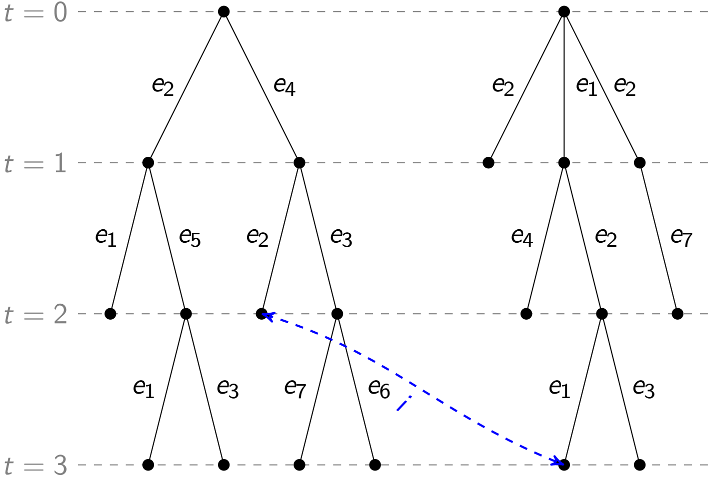
Parameters of the Models

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both? Branching time or linear time?

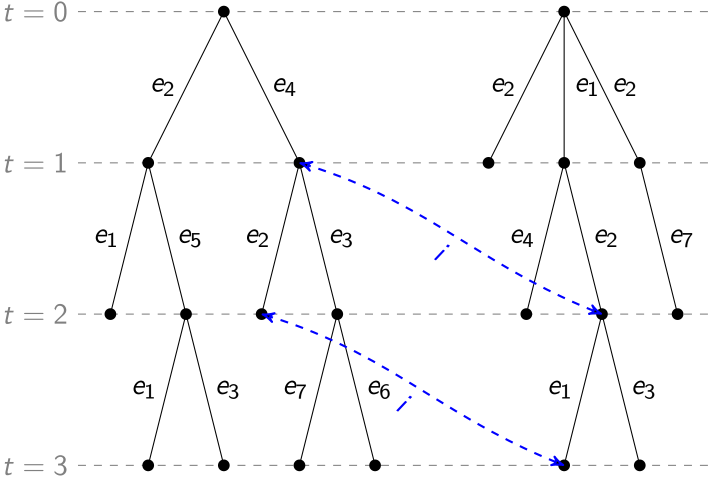
2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

3. Conditions on the reasoning abilities of the agents. Do the agents satisfy perfect recall? No miracles? Synchronization?

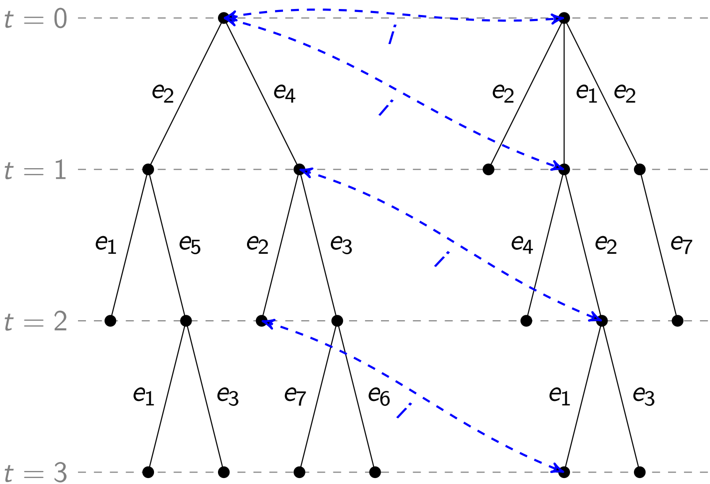
Perfect Recall



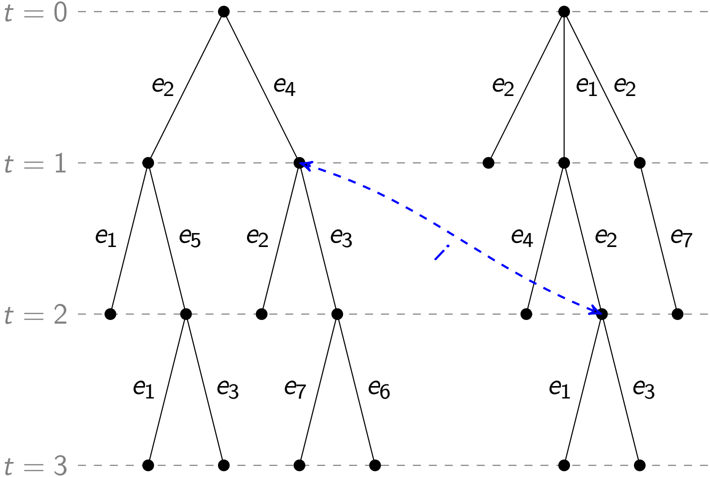
Perfect Recall



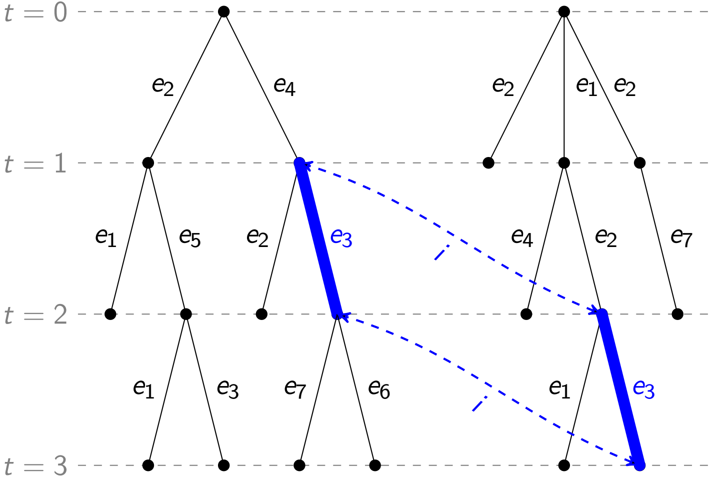
Perfect Recall



No Miracles



No Miracles



Theorem(s) Expressive language plus certain idealizations can lead to high complexity (Π^1_1-completeness).

J. Halpern and M. Vardi. *The Complexity of Reasoning abut Knowledge and Time*. *J. Computer and Systems Sciences*, 38, 1989.

See also,

J. van Benthem and E. Pacuit. *The Tree of Knowledge in Action: Towards a Common Perspective*. AiML 2006.

✓ Introductory Example

✓ Motivating Examples

✓ Background: History Based Models

4. Knowledge Based Obligations.

   E. Pacuit, R. Parikh and E. Cogan. *The Logic of Knowledge Based Applications*. Knowledge, Rationality and Action (Synthese) 149: 311 - 341 (2006).

5. Discussion

Actions

Assume a finite set, $Act \subseteq \Sigma$, of primitive actions.

Assume $Act = \cup_{i \in \mathcal{A}} C_i$ where $C_i \cap C_j \neq \emptyset$ for $i \neq j$.

Actions

Assume a finite set, $Act \subseteq \Sigma$, of primitive actions.

Assume $Act = \cup_{i \in \mathcal{A}} C_i$ where $C_i \cap C_j \neq \emptyset$ for $i \neq j$.

Given a finite global history H and $a \in Act$,

$$a(H) = \{H' \mid Ha \preceq H' \text{and } H' \text{a global history}\}$$

Actions

Assume a finite set, $Act \subseteq \Sigma$, of primitive actions.

Assume $Act = \cup_{i \in \mathcal{A}} C_i$ where $C_i \cap C_j \neq \emptyset$ for $i \neq j$.

Given a finite global history H and $a \in Act$,

$$a(H) = \{H' \mid Ha \preceq H' \text{and } H' \text{a global history}\}$$

$$H, t \models [a]\varphi \;\text{ iff }\; H', t+1 \models \varphi \;\text{ for each }\; H' \in a(H)$$

Actions

$$H, t \models [a]\varphi \text{ iff } H', t+1 \models \varphi \text{ for each } H' \in a(H)$$

Actions

$$H, t \models [a]\varphi \;\text{ iff }\; H', t+1 \models \varphi \;\text{ for each }\; H' \in a(H)$$

- When an action is performed, it is performed at the next moment of time.

Actions

$$H, t \models [a]\varphi \text{ iff } H', t+1 \models \varphi \text{ for each } H' \in a(H)$$

- When an action is performed, it is performed at the next moment of time.
- Only one agent can perform some action at any moment.

Actions

$$H, t \models [a]\varphi \;\text{ iff }\; H', t+1 \models \varphi \;\text{ for each }\; H' \in a(H)$$

- When an action is performed, it is performed at the next moment of time.
- Only one agent can perform some action at any moment.
- If no agents perform an action, then nature performs a 'clock tick'.

Actions

$$H, t \models [a]\varphi \text{ iff } H', t+1 \models \varphi \text{ for each } H' \in a(H)$$

- When an action is performed, it is performed at the next moment of time.
- Only one agent can perform some action at any moment.
- If no agents perform an action, then nature performs a 'clock tick'.
- Each agent knows *when* it can perform an action. $(\langle a_i \rangle \top \rightarrow K_i \langle a_i \rangle \top)$

Values: Informal Definition

All global histories will be presumed to have a value

Let $\mathcal{G}(H)$ be the set of extensions of (finite history) H which have the highest possible value. (Assumptions are needed to make $\mathcal{G}(H)$ well defined)

Values: Informal Definition

All global histories will be presumed to have a value

Let $\mathcal{G}(H)$ be the set of extensions of (finite history) H which have the highest possible value. (Assumptions are needed to make $\mathcal{G}(H)$ well defined)

We will say that a is good to be performed at H if $\mathcal{G}(H) \subseteq a(H)$, i.e., there are no H-good histories which do not involve the performing of a.

Values: Formal Definition

Let $\mathcal{H}$ be a set of global histories and $H \in \mathcal{H}$ a global history. For each $t \in \mathbb{N}$, let $\mathcal{F}(H_t) = \{H' \in \mathcal{H} \mid H_t \preceq H'\}$.

A value function is a map $\text{val} : \inf(\mathcal{H}) \rightarrow \mathbb{R}$ such that

1. For all $t \in \mathbb{N}$, $val[\mathcal{F}(H_t)]$ is a closed and bounded subset of $\mathbb{R}$.
2. $\cap_{t\in\mathbb{N}} val[\mathcal{F}(H_t)] = \{val(H)\}$

Define $\mathcal{G}(H_t) = \{H' \mid H' \in \mathcal{F}(H_t) \text{ and } \text{val}(H') = \max(\text{val}[\mathcal{F}(H_t)])$

Knowledge Based Obligation

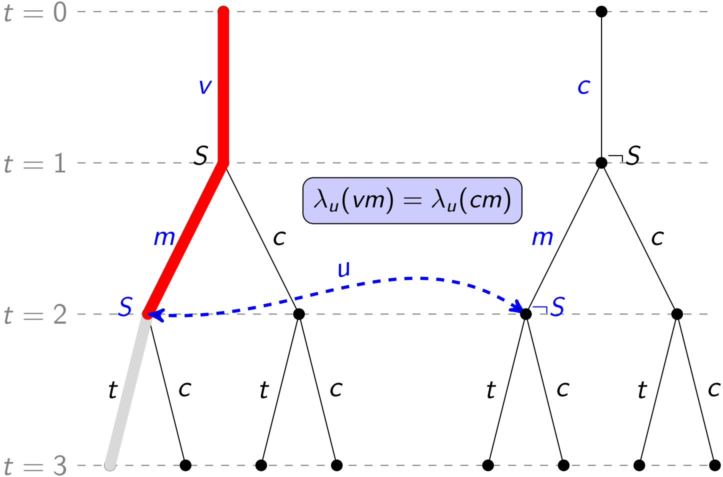
Agent i has a (knowledge based) obliged to perform action a at global history H and time t iff a is an action which i (only) can perform, and i *knows* that it is good to perform a.

For each $a \in \text{Act}$, let $G(a)$ be a formula:

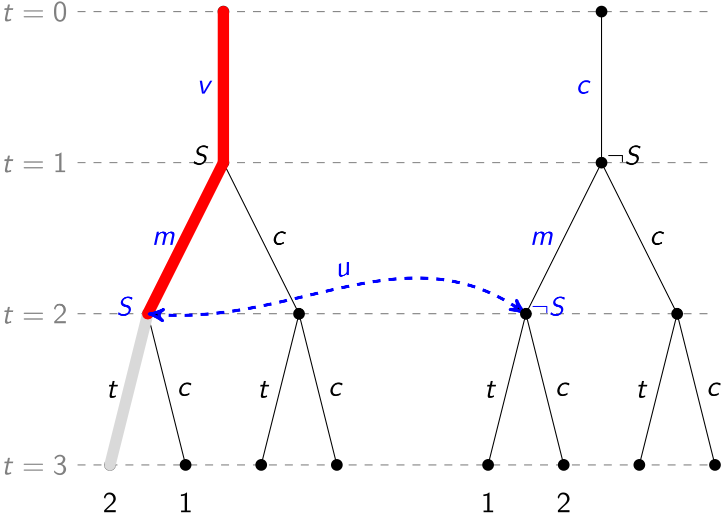
$$H, t \models G(a) \text{ iff } \mathcal{G}(H_t) \subseteq a(H_t)$$

Then we say that i is obliged to perform action a (at H, t) if $K_i(G(a))$ is true (at H, t).

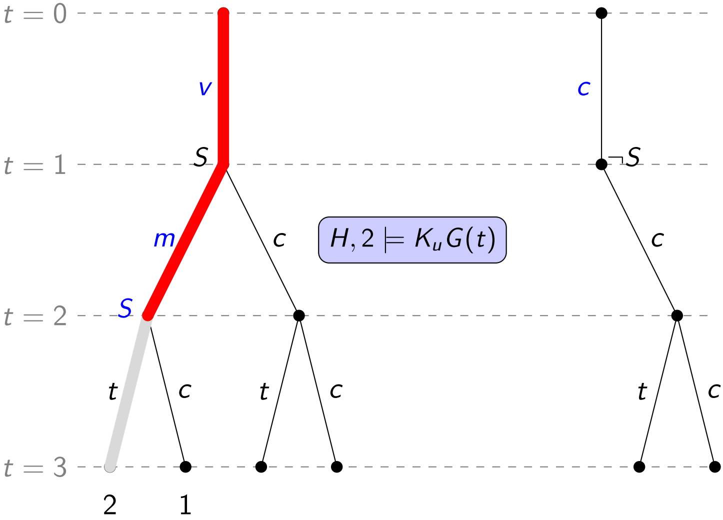
Example 2



Example 2



Example 2



Default Obligations (Sketch)

We introduce a modal operator B_i which is intended to mean that "i has a default belief that ....".

Assume a set of Grove spheres on the set $\mathcal{H}$: $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots\}$ where for each $i \geq 1$, $\mathcal{S}_i \subseteq \mathcal{S}_{i+1} \subseteq \mathcal{H}$, and $\cup_{i=1}^{\infty} \mathcal{S}_i = \mathcal{H}$.

Let H be a finite global history, $\mathcal{D}(H)$ is the set of the most plausible histories which extend H.

$\mathcal{D}_i(h)$: the set of histories that i considers plausible.

Default Obligations (Sketch)

$$H, t \models B_i\varphi \text{ iff for all } H', H'_t \in \mathcal{D}_i(\lambda_i(H_t)),\ H', t \models \varphi$$

$K_i\varphi$ semantically entails $B_i\varphi$.

B_i is a **KD45** operator.

$B_iG(a)$: "agent i has a default knowledge based obligation to perform a"

Our Language

- $G(a)$: "a is a non-informational obligation".
- $\langle a \rangle \top$: "action *a can* be performed".
- $K_i \langle a_i \rangle \top$: "agent i knows that she can perform action a_i".
- $K_i G(a_i)$: "i has a (knowledge based) obligation to perform a_i".
- $B_i \varphi$: "agent i (justifiably) believes φ".
- $B_i G(a_i)$: "agent i has a default obligation to perform a_i".

✓ Introductory Example

✓ Motivating Examples

✓ Background: History Based Models

✓ Knowledge Based Obligations.

> E. Pacuit, R. Parikh and E. Cogan. *The Logic of Knowledge Based Applications*. Knowledge, Rationality and Action (Synthese) 149: 311 - 341 (2006).

5. Discussion

Recall that Ann has the (knowledge based) obligation to tell Jill about her father's illness ($K_a G(m)$).

Recall that Ann has the (knowledge based) obligation to tell Jill about her father's illness ($K_a G(m)$).

Clearly, Ann will not be under any obligation to tell Jill that her father is ill, if Ann justifiably believes that Jill would not treat her father even if she knew of his illness.

Recall that Ann has the (knowledge based) obligation to tell Jill about her father's illness ($K_a G(m)$).

Clearly, Ann will not be under any obligation to tell Jill that her father is ill, if Ann justifiably believes that Jill would not treat her father even if she knew of his illness.

Thus, to carry out a deduction we will need to assume

$$K_j(K_u \text{sick} \leftrightarrow \bigcirc \text{treat})$$

A similar assumption is needed to derive that Jill has an obligation to treat Sam.

A similar assumption is needed to derive that Jill has an obligation to treat Sam.

Obviously, if Jill has a good reason to believe that Ann always lies about her father being ill, then she is under no obligation to treat Sam.

A similar assumption is needed to derive that Jill has an obligation to treat Sam.

Obviously, if Jill has a good reason to believe that Ann always lies about her father being ill, then she is under no obligation to treat Sam.

In other words, we need to assume

$$K_j(\text{msg} \leftrightarrow \text{sick})$$

Common Knowledge of Ethicality

These formulas can all be derived for one common assumption which we call *Common Knowledge of Ethicality*.

Common Knowledge of Ethicality

These formulas can all be derived for one common assumption which we call *Common Knowledge of Ethicality*.

1. The agents must (commonly) know the protocol.
2. The agents are all of the same "type" (social utility maximizers)

What is a Protocol?

Given the full tree T of events, a **protocol** is *any* subtree of T.

What is a Protocol?

Given the full tree T of events, a **protocol** is *any* subtree of T.

- A protocol is the set of histories compatible with some **process**, i.e., it is the "unwinding" of a (multi-agent) state transition system.

What is a Protocol?

Given the full tree T of events, a **protocol** is *any* subtree of T.

- A protocol is the set of histories compatible with some **process**, i.e., it is the "unwinding" of a (multi-agent) state transition system.
- A protocol is the set of histories satisfying some **property**:
  - Physical properties: every message is eventually answered, no message is received before it is sent
  - Agent types: agent i is the **type** of agent who always lies, agent j is the type who always tells the truth

What is a Protocol?

Given the full tree T of events, a **protocol** is *any* subtree of T.

- A protocol is the set of histories compatible with some **process**, i.e., it is the "unwinding" of a (multi-agent) state transition system.
- A protocol is the set of histories satisfying some **property**:
  - Physical properties: every message is eventually answered, no message is received before it is sent
  - Agent types: agent i is the **type** of agent who always lies, agent j is the type who always tells the truth
- A protocol is the set of histories of an extensive game consistent with a **strategy profile.**

Defining a Protocol

Defining a Protocol

1. What formal language should we use to define the protocol?
2. What models do we have in mind?

Defining a Protocol

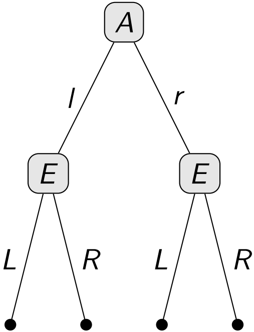
1. What formal language should we use to define the protocol?
2. What models do we have in mind?

Given a formula φ, two ways to think about defining a protocol:

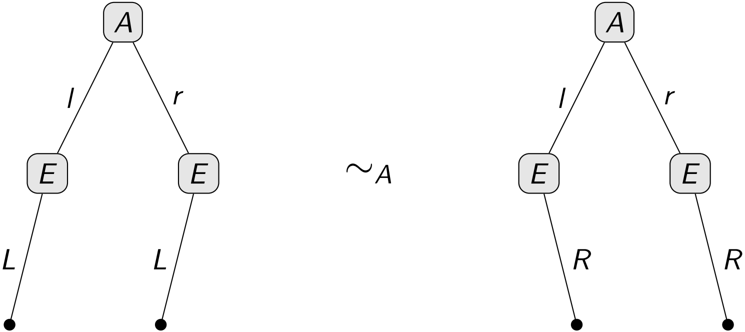
Set of histories: the set of histories P in the full event tree T such that $h \in P$ iff $h \models \varphi$

Set of models: the set $\mathrm{Mod}(\varphi)$ (the set of models of φ)

Knowledge of the Protocol



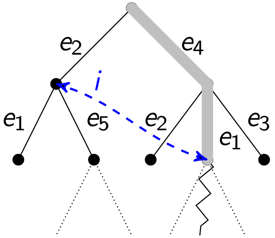
Knowledge of the Protocol



Two types of uncertainty?

Given two finite histories h and h',

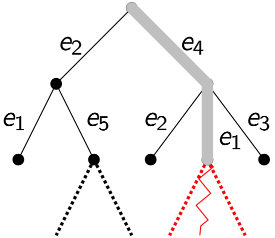
> $h \sim_i h'$ *means* given the events i has observed, *h and h' are indistinguishable*



Two types of uncertainty?

Given two **maximal histories** H and H',

agent i may be uncertain which of the two will be the final outcome.



Thank You!